WORKING WITH DIGITAL FILTERS

by Peter Best

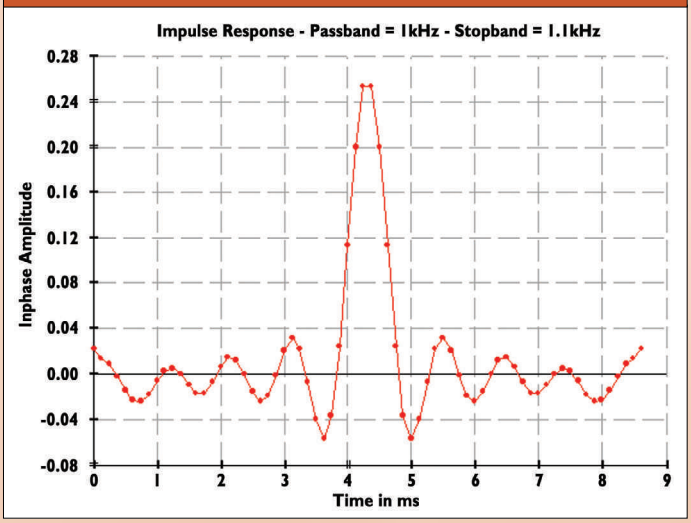
Most of the books and technical papers that describe digital filtering consist mostly of complex mathematical concepts with little to no emphasis placed on the practical implementation of a physical digital filter. The math behind digital filtering techniques is indeed interesting, but you don't have to be a mathematician to design and build a working digital filter. Keep reading and I'll prove it to you.

Poles and Zeros

Digital filter theory and the math that accompanies it describe the process of placing a number of poles and zeros at strategic locations in complex mathematical planes to obtain a desired filter response. For filter designs that are realized in the mathematical S plane, the filter response is called the system function (Hs). The system function is actually based on the division of two sets of complex polynomials.

Mathematically, the poles of a filter are the roots of the system function's complex polynomial denominator and the filter's zeros are the roots of the system function's complex polynomial numerator; this results in the system function Hs. The poles and zeros are represented physically by resistors, capacitors, and opamps. Digitally, poles and zeros and their placement in a plane are represented by firmware-resident coefficients and a filter kernel.

Figure 1. This low-pass filter impulse response plot contains 70 points or taps that we could use as coefficients in a digital filter algorithm.



A filter kernel is really the impulse response of the filter. Mathematically, an impulse is a spike of sorts that occurs at time = 0 with infinite positive amplitude. In addition, a mathematical impulse has no width and a total area that is equal to 1. Applying an impulse to a filter — which excites the filter equally at all frequencies — results in a response from the filter — which is the filter's impulse response. We care about impulse response because a filter's impulse response plot contains the points (the filter kernel) we use for coefficients that are used in our digital filter firmware algorithm. A typical low-pass filter kernel or impulse response plot can be seen in Figure 1.

The best way to describe poles and zeros is to imagine a square yard of very flexible rubber stretched horizontally, one inch above a piece of plywood. The stretched rubber is the mathematical plane S. A zero is formed in the plane by placing a thumbtack through the rubber, into the plywood. The tack forms a depression in the rubber that is shaped like a funnel cloud. This is a zero and mathematically it extends infinitely negative.

Frequencies in the plane that are close to zero points are attenuated. If you placed two tacks (zeros) six inches apart and pulled up the rubber in the center between the tacks, you will create a pole between the two zeroes. Frequencies near the poles in the planes are amplified. Mathematically, a pole extends infinitely positive. Imagine being able to tack down areas and pull up areas of the rubber plane to adjust the attenuation and amplification of certain frequency areas. That's what poles and zeros do.

Filter order is determined by the number of poles the filter contains. For instance, a filter with two poles is termed a second order filter. More poles and zeros mean more code; thus, more time will be needed to evaluate the filter kernel and incoming data samples in a digital filter. For an analog filter, the number of poles and zeros determines how much hardware you're going to be assembling. The more poles and zeros that are included in the analog filter design, the more resistors, capacitors, and opamps you'll have to add to create them.

The tradeoff for an increased number of poles and zeros (filter order) for both digital and analog filters is better filter response. For instance, a simple RC low-pass filter (Figure 2a) contains a single pole. The Sallen-Key filter configuration in Figure 3 has two poles. Because filter gain and filter output impedance can be controlled, the Sallen-Key active filter will perform better than the

simple passive RC filter network. A second pole can be added to the RC filter by simply adding a second RC filter at the output of the first RC filter, as shown in Figure 2b.

Since there are no zeros in either the Sallen-Key or the RC filters I've described, they are called all pole filters. Mathematically, the numerator of the system function Hs for the RC and Sallen-Key all pole filters is 1, as all of the work is done with poles, the roots of which are found in the denominator of the system function.

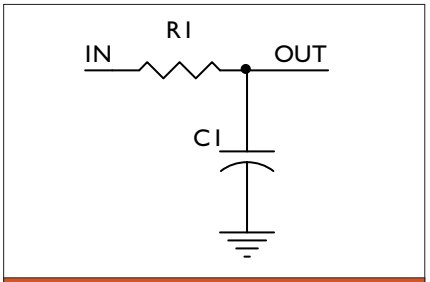


Figure 2a. If an impulse is applied to this circuit, the time taken for the output to ramp up to the input voltage is determined by the size of the resistor. The higher the resistance, the longer it takes to charge the capacitor, thus, the slower the ramp up process.

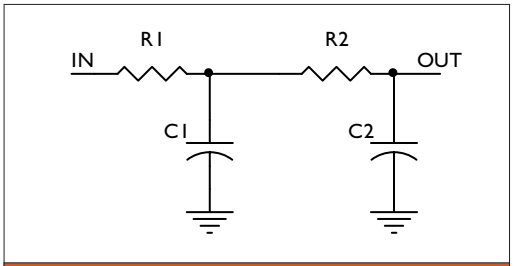


Figure 2b. You can only do this up to a point. Every additional RC stage adds additional loading on the output side of the filter with minimal payback. If you must cascade filter elements, it is better to use active opamp filters that can be easily made to control filter gain and filter output loading while adding more responsiveness to the filter.

All of this pole and zero talk is important, as poles and zeros determine how a filter responds and we, as humans, have names and generic S-plane pole-zero plots we can tie to those generic filter responses. We won't be plotting pole and zero functions during the course of our digital filter design, as it isn't a necessary process in the generation of our digital filter. Instead, we'll use Microchip's Filter Lab application to put the poles and zeros in the correct places and compute the values of our Sallen-Key filter components. If you want to take a look at some pole-zero plots, one of the best application notes I've seen that illustrates the pole and zero concept with actual three-dimensional color plots is the Dallas/Maxim application note APP 733.

Filter Basics

There are three common filter types that are used to describe the behavior of both digital and analog filters. The three filter models — or approximations — are called Chebyshev, Bessel, and Butterworth. You'll sometimes see Chebyshev spelled in many other ways, including Tschebyshceff. To understand the characteristics of each of the filter types, you must first understand the terminology used to describe a typical digital or analog filter.

A filter will pass signals at their maximum amplitude in an area known as the filter's passband. Conversely, a filter will severely attenuate and block signals in its stopband area. A high-pass filter passes frequencies above its cut-off frequency and blocks them below its cut-off frequency. A low-pass filter passes signals below its designed cut-off frequency and blocks them above the cut-off frequency. A bandpass filter passes signals that are inside a particular frequency range and a bandstop filter blocks signals in a frequency range. All of these filter types have one thing in common. They can all be derived from a low-pass filter design.

A high-pass filter is the spectral inversion of a low-pass filter. To spectrally invert a filter kernel, you only need to apply opposite signs to each filter kernel point and add 1 to the center filter kernel sample. I used an application called ScopeFIR to generate the low-pass filter kernel in Figure 1 and its spectrally inverted high-pass kernel in Figure 4. The physical low-pass filter configurations in Figures 2 and 3 can be converted to high-pass filter configurations by simply swapping the locations of the resistors and capacitors. Now that you've generated a high-pass filter from a low-pass filter kernel, you can take it a step further and add the low-pass kernel and the high-pass kernel you just generated to form a bandstop filter.

No matter which filter model you choose, the ideal filter would have absolutely no frequency gap between the passband and stopband. This no-gap scenario is called a brickwall response, which is envisioned as a theoretically perfect filter with a vertical drop of the attenuation from the passband to the stopband. In a physical filter, there will be some gap between the passband and the stopband. This gap is called the transition band. The width of the gap is

Hobbyists, Designers, Students:

Why wire-wrap when you can get PCB prototypes for much less?

Our Standard PCBs include Tooling, Mask, and Legend

2 Layer 5 day $13 each

4 Layer 5 day $33 each

Our "Bare PCBs" have no mask and no legend but can give you even more SAVINGS!

$40 lot + 60cents per square inch for 2 layer Bare PCBs

Easy Online Ordering at:

www.PCBFABExpress.com

PCB FABEXPRESS

High Quality PCBs @ Low Impact Prices

Email: support@pcbfabexpress.com Tel: (408) 857 0039

Circle #112 on the Reader Service Card.

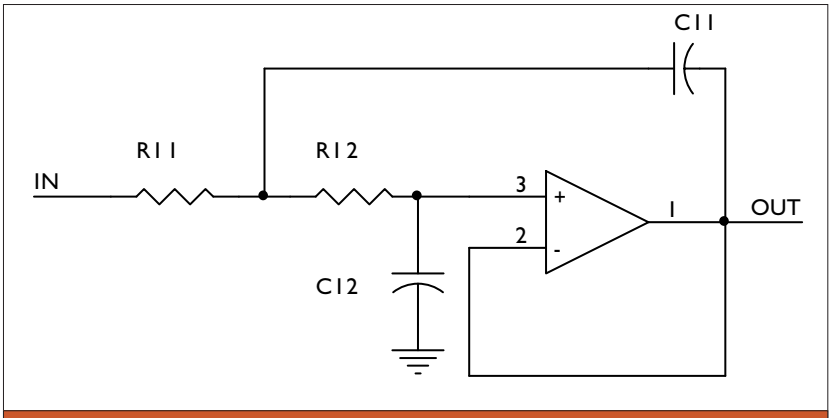


Figure 3. This circuit can be used to implement Chebyshev, Butterworth, and Bessel analog filters by simply changing the values of the components.

referred to as the transition bandwidth. As the transition bandwidth gets smaller, the filter's frequency response improves accordingly. Transition bandwidth is inversely proportional to the number of poles in the filter.

Figure 5 shows a filter design that has a passband of 1 kHz and a stopband of 4 kHz with a stopband attenuation of -30 dB. Note that the ideal passband cutoff of 1 kHz doesn't really occur until sometime beyond 2 kHz. You can safely assume that this filter design won't require many components and won't take up much space in a microcontroller's memory area.

Judging from the transition bandwidth, you can also assume that this filter will not be very responsive to small changes in frequency. By adding some poles and zeros and specifying a lower stopband frequency, Figure 6 shows that the width of the transition band narrows and sharpens the roll-off at the start of the transition band. This filter design requires quite a bit more firmware and hardware than the filter design depicted in Figure 5 and is much more responsive to small changes in frequency.

Filter Comparisons

The Chebyshev filter has a sharper roll-off than the Butterworth and Bessel filter models. The Chebyshev filter uses ripple in the passband to achieve the sharp roll-off characteristic. The more ripple allowed in the Chebyshev passband, the sharper the roll-off that can be obtained from the filter. Although the Chebyshev filter does a great job at discriminating frequencies, it has a relatively poor step response when compared to a Bessel filter.

Poor step response means that the Chebyshev does a poor job of filtering signals with sharply rising and falling edges. The Chebyshev filter will overshoot and undershoot on the rising and falling edges of the input signal. You're already familiar with this phenomenon: It's called ringing. The Chebyshev filter is composed of poles only. A Filter Lab-generated Chebyshev frequency response plot is shown in Figure 7.

Ringing on the rising and falling signal edges can be eliminated by employing a Bessel filter model. Unlike the Chebyshev filter, the Bessel filter does not contain any ripple in the passband or the stopband. However, the Bessel filter's roll-off rate of attenuation in the transition band is the worst of the three filter models. The Bessel filter shines with sharp-edged signals as it introduces very little overshoot or ringing and preserves the majority of the sharpness of the filtered signal's edges. Thus, the Bessel filter model is a good choice for filtering time domain signals. The Bessel filter model in Figure 8 contains no zeros and is an all poll filter.

As you can see in Figure 9, a Butterworth filter is maximally flat in the magnitude response in the filter's passband with no ringing in the stopband. Like the Chebyshev, the

ATOMIC TIME
1010 Jorie Blvd. #332
Oak Brook, Il 60523
1-800-985-8463
www.atomictime.com

H15U
SALE!
Office School Clock #1
WT-3121A $39.95>
This wall clock is great for an office, school, or home. It has a professional look, along with professional reliability. Features a manual set option, daylight saving time disable option, and a safe plastic lens and case.
Atomic Digital Wristwatch
< H15U $34.95
A high tech digital wristwatch with a sophisticated look. Features a metal 'stretch' band and a high-contrast digital display. 12/24 hr time formats, backlight, date, and day of week.
WT-3121A
56G24-4
Arcron Atomic Watch
<56G24-4 $249.99
This elegant watch features a shock-resistant titanium case with hardened mineral lens. Silver dial with arabic numerals, and high quality replaceable leather band. Watch can change to any world time zone. Case diameter 40mm. Made in Germany.
1-800-985-8463
www.atomictime.com
WS-8007U-C
^LaCrosse Digital Wall Clock $34.95
This digital wall / desk clock comes with a beautiful cherry wood frame. It shows time, date, day of week, temperature and moon phase. 12/24 format.
Tell time by the U.S. Atomic Clock -The official U.S. time that governs ship movements, radio stations, space flights, and warplanes. With small radio receivers hidden inside our timepieces, they automatically syncronize to the U.S. Atomic Clock (which measures each second of time as 9,192,631,770 vibrations of a cesium 133 atom in a vacuum) and give time which is accurate to approx. 1 second every million years. Our timepieces even account automatically for daylight saving time, leap years, and leap seconds. $7.95 Shipping & Handling via UPS. (Rush available at additional cost) Call M-F 9-5 CST for our free catalog.
Circle #137 on the Reader Service Card.

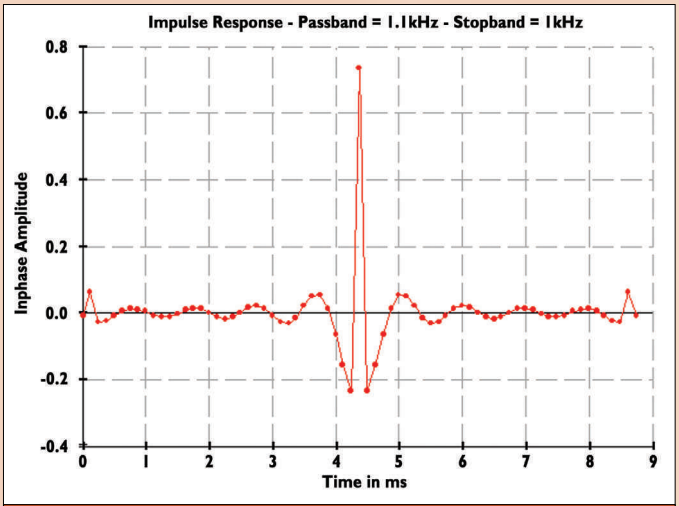


Figure 4. This says it all. The top sine wave's amplitude decreased as I increased the input frequency towards the digital filter's 1 kHz bandstop value.

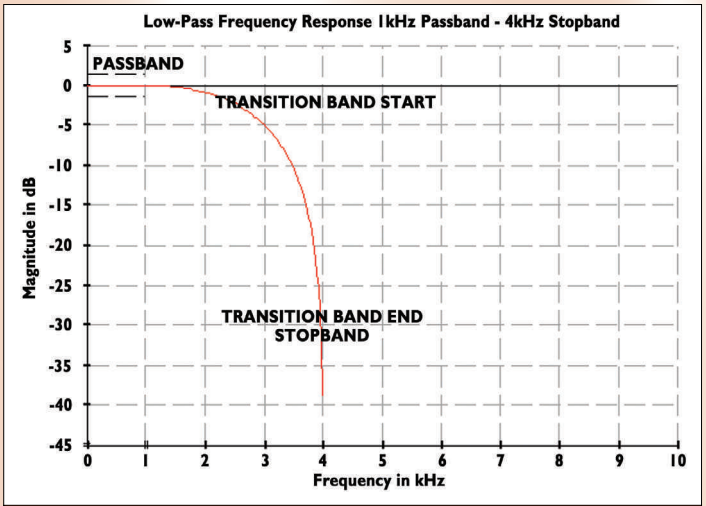


Figure 5. A relatively slow change in attenuation over the transition band translates into a filter that doesn't require as many poles and zeros and, thus, isn't as responsive as the filter shown in Figure 6.

Butterworth filter is good for separating bands of frequencies. The Butterworth filter stands in the middle as far as transition band attenuation is concerned. Butterworth's filter transition band attenuation is better than the Bessel's, but worse than the Chebyshev's. The step response of a Butterworth filter is slightly better than the Chebyshev step response. There are no zeros in the transfer function of a Butterworth filter. Thus, like the Chebyshev and Bessel filter models, the Butterworth filter is an all pole filter.

Butterworth and Chebyshev filter models work best in the frequency domain, while the Bessel is a better time domain signal filter. It's easy to determine if you're looking

Surplus Sales of Nebraska

Weston 1751 Series Meters

(MTR) 1751-62
0 - 100% Cam Dwell
0 - 200 uadc full scale.
Needle reversed.

(MTR) 1751-63
Vacuum Pressure
500 - 0 - 500µ amps
DC full scale.

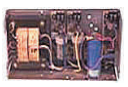
(MTR) 1751-65
0 - 10 / 0 - 50 RPM
in 100's. 1 madc full scale.

(MTR) 1751-67
0 - 10 / 0 - 50 amps DC
1 madc full scale.

(MTR) 1751-69
0 - 10 / 0 - 20 vdc
1 madc full scale.

(MTR) 1751-91
0 - 50 kvdc.
200 µa full scale.
Req. bleeder resistor.

$22 each - 5 or more for $20 each - Mix & Match
**** Screen Your Own Meters!! Blank White Meter Faces available!! $2 each ****



(PS) 10656X
Yes, we still have a good supply of this new, wonderful Deltron triple voltage open frame linear supply. So versatile. Trim the 15 vdc fundamental supplies on the circuit board. 12-16v or remote mount a 10 turn 5k Ohm pot and turn it into a 7-30v regulated lab supply. It's simple to do. Instructions included. Series or parallel the two supplies for a combined 7-60v @ 2.5-5 amps. Board trimmable current limiting too. New, boxed, from Collins surplus. 10 pounds each. 3-1/4" x 5" x 11" long.

$35 each - $32 (3+)

Please visit our website with over 2,000 pages and 10,000 images • www.surplussales.com

HI Manuals: 1,000s of radio, military and test equipment manuals on hand!!

Fans & Blowers: Over 100,000 In Stock! Muffin - Squirrel - Spiral

Transformers: Sealed, Open Frame, Audio, Switching, Pulse, High Voltage

Resistors: Wire Wound, Carbon Comp, Metal Film, Surface Mount

Split Beads: For are all of your interference problems.

Ceramic Transmitting Capacitors: Of All Sizes: Doorknobs-Feedthru-Water Cooled

Metal Halide High Intensity Lamp Ballasts

Kapton, Fiberglass and Mylar Tapes

Vacuum Tubes: Over 1,000,000 vacuum tubes in stock!!

Enclosures: Rack Cabinets, Builder Boxes, Shipping Containers

Antique Galvanometers: by JH Bunnell

Heatsinks: All Sizes and Shapes. Large Quantities for Manufacturers

Western Union Self-Winding Clocks

Toroids: Powdered Iron, Ferrite, Rods, Pot and E Cores

Variable Capacitors: Air - Mica - Vacuum

1502 Jones Street, Omaha, NE 68102 • Fax: 402-346-2939 • e-mail: grinnell @ surplussales.com • Visa, Mastercard, American Express or Discover
Call or e-mail for shipping and total charges. All SPECIALS in this Ad only good for 30 day advertising cycle. No exceptions please.

Visit our website @
www.surplussales.com

800-244-4567 • 402-346-4750

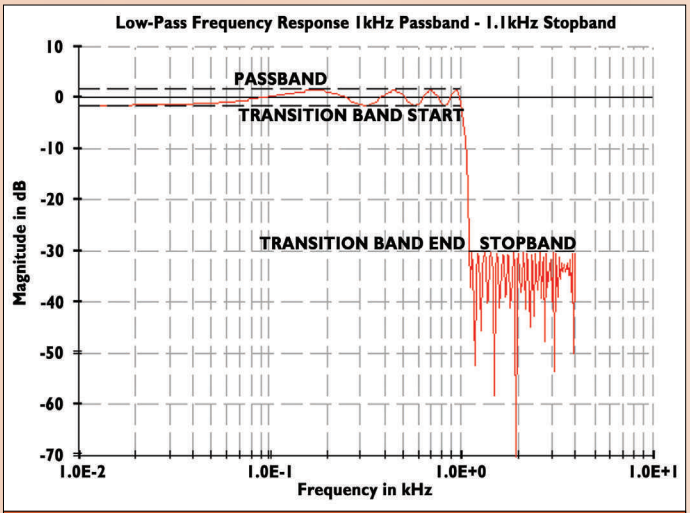


Figure 6. Due to the tighter transition bandwidth, the frequency attenuation in the transition band is much sharper, which equals more poles and zeros and a more refined filter.

at a time or frequency domain filter plot. If the X axis is measured in units of time, you're in the time domain and the information is carried as a function of time. If the X axis is measured in units of frequency, you're operating in the frequency domain, in which the signal's information is carried as a function of frequency. Don't let the logarithmic scales throw you. The information contained in a log plot is the same information that would exist in a unit plot of the same data. Log scales are used when standard unit scales can't dynamically give your eyes the full impact of the plot.

You're all set on basic filter theory as now you know that, to design an analog low-pass filter, you will need to specify the passband frequency, the stopband frequency, and the stopband attenuation. The same is true for a digital low-pass filter design. You also know that, once you design a low-pass filter, you can derive any other filter type from your low-pass filter kernel. You've also been introduced to two applications: ScopeFIR and Filter Lab, which help take the tedious math out of designing filters. In the text that follows, we will design and build some analog and digital filters using ScopeFIR and Filter Lab.

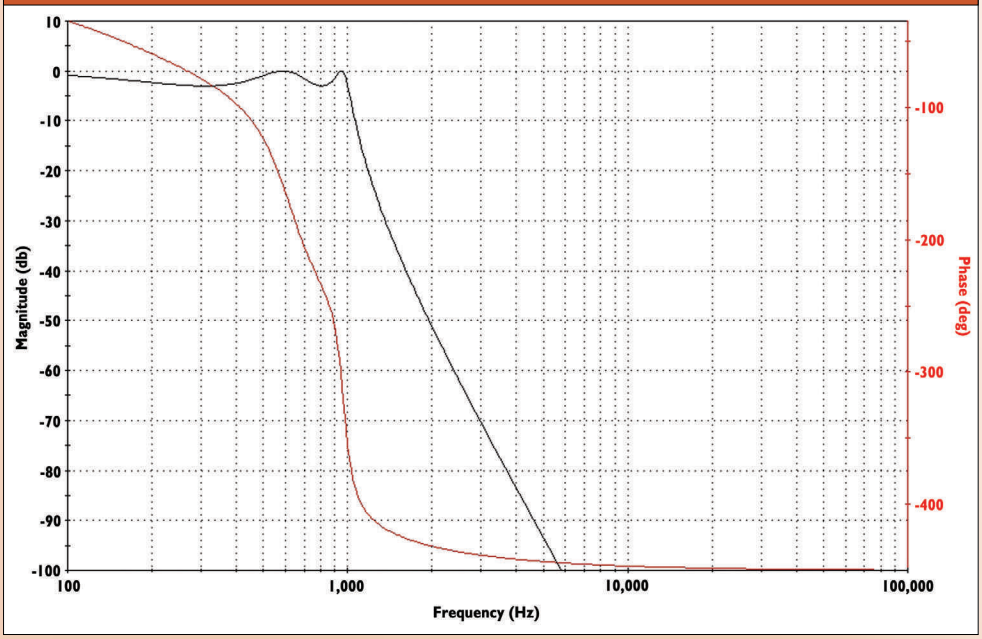
The Digital Filter Hardware

Digital filters have many advantages over analog filters. To change the characteristics of a digital filter, all you have to do is change some code versus altering physical components in an analog filter. Another advantage of using a digital filter is that there are no physical components that can drift in value and degrade the performance of your filter. Even with those extras, a good digital filter is incomplete without an analog filter counterpart. This sidekick analog filter is known as an anti-aliasing filter.

Our digital filter input is gathered by the analog-to-digital converter module in a PIC18F452. The maximum frequency that our PIC-based digital filter can resolve is set to one half of the sampling frequency, which we will set to 8 kHz. If we attempt to sample a signal that has a frequency component above one half of our sampling rate, we will record an aliased signal, which will not represent the true input signal.

Aliasing occurs when signal components above one half of the sampling rate are folded back and appear as lower frequencies in the digital output. It's the anti-aliasing filter's job to block any signals above one half of the sampling frequency and pass all other signals below one half of the sampling frequency to the PIC18F452's analog-to-digital converter unchecked. The one half of the sampling frequency value is known as the Nyquist frequency or Nyquist rate.

Figure 7. By using ripple in the passband, the Chebyshev sacrifices response in the time domain to optimize response in the frequency domain.



If you check out the schematic (Figure 13), you'll see that I've devised a digital filter circuit using a PIC18F452 that includes a Microchip MCP604 quad opamp, a couple of Microchip MCP42100 dual digital potentiometers, a +5 V DC regulated power supply, a Sipex SP233ACP-based serial port, a Microchip ICSP programming port, and a 40 MHz microcontroller clock. My prototype digital filter board can be seen in Figure 11.

A typical digital filter configuration consists of an anti-aliasing filter feeding an analog-to-digital converter with the output of the digital filter flowing into a digital-to-analog converter. A pair of opamps in

the MCP604 handles the anti-aliasing and digital-to-analog converter duties with both opamps being supported by the full complement of MCP42100 100K Ω digital potentiometers. Along with a quad of capacitors, the MCP604 and MCP42100 form two Sallen-Key filter configurations that are dynamically controlled via the digital potentiometers, which are adjusted using a personal computer-resident control program. Since every digital pot has a different end-to-end (PA-to-PB) resistance, the digital filter control panel has dials to set the actual digital potentiometer PA-to-PB resistance so that each click of the potentiometer's virtual dial displays the correct resistance in the digital filter control panel's virtual LED displays.

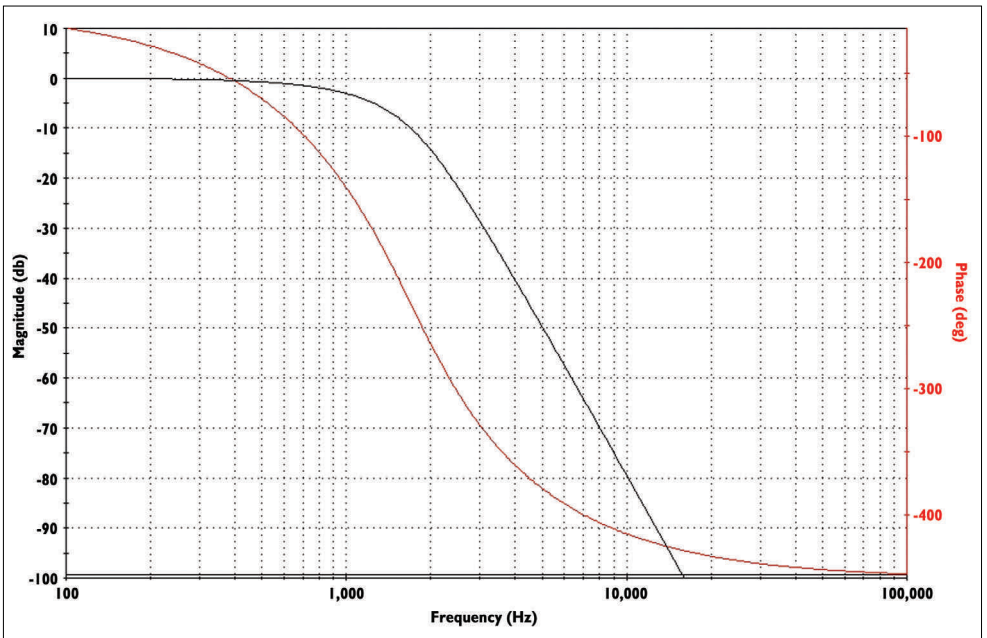


Figure 8. The Bessel filter model produces a constant time delay with respect to frequency. Note the symmetrical phase plot. Another feature of the Bessel filter is its ability to deliver linear phase performance over a wide frequency range. That's why the Bessel filter has very good pulse (step) response.

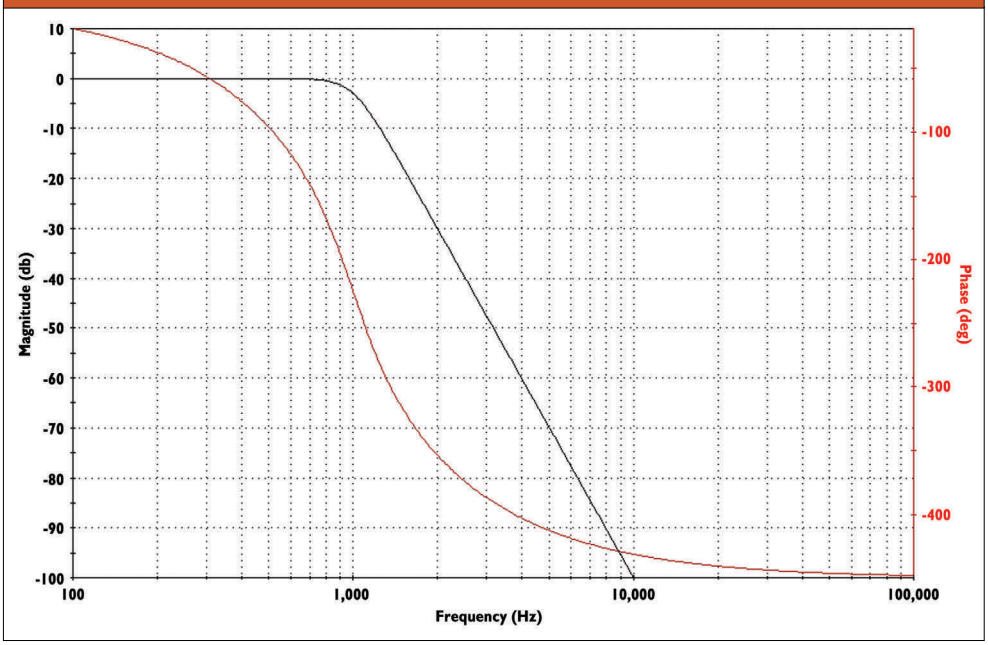
Microchip's Filter Lab program is used to determine the Sallen-Key analog filter resistances for a particular filter type (Chebyshev, Bessel, or Butterworth) and cut-off frequency and the computed resistances are transferred to the digital potentiometers using the digital filter control panel's virtual dials and a serial connection between the personal computer and the digital filter electronics.

There is no practical way to adjust the capacitors in the analog filter circuits. So, fixed capacitor values were chosen that provide a wide range of analog filter bandwidth when combined with the 100K Ω digital potentiometers. The digital filter control panel also allows remote control of the PIC18F452's PWM (Pulse Width Modulation) module. The digital filter control panel — along with the opamps, fixed capacitors, and digital potentiometers — allows us to create any of the three basic filter types (Chebyshev, Bessel, or Butterworth) and specify the analog filter response without having to swap physical components. To add to that flexibility, the opamp is not committed to a printed circuit board land pattern. This allows the user to configure the quad of opamps in any way desired.

There's nothing unusual about the circuitry surrounding the PIC18F452. A standard LM340S +5 V DC regulated power supply circuit is included onboard the digital filter printed circuit board. The LED I used as the power indicator has an internal current limiting resistor and sits directly across the +5 V DC output of the voltage regulator.

To establish a communications session between the digital filter board and the personal computer, the USART receive and transmit pins of the PIC18F452 are connected

Figure 9. It takes a higher order Bessel filter to be as responsive as a typical Butterworth filter. However, the trade-off is worth the extra firmware in a digital filter and the extra hardware in ananalog filter design.



in a standard way to the Sipex SP233ACP RS-232 converter, which, in turn, interfaces to the outside world via a standard nine-pin shell connector as a DCE device. By making the digital filter board's RS-232 port DCE, the need for an RS-232 crossover cable is eliminated.

The SPI module of the PIC18F452's MSSP (Master Synchronous Serial Port) drives the digital potentiometers and a 40 MHz crystal oscillator provides the clocking for 100 nS instruction cycles within the PIC18F452. A standard Microchip ICSP port is also a part of the digital filter board and allows the use of the MPLAB ICD2 hockey puck for programming and debugging.

You can easily assemble the digital filter electronics using point-to-point techniques. If you go the point-to-point route, you can substitute a DIP version of the Sipex SP233ACP for the SMT version on the digital filter printed circuit board. For those of you that prefer a printed circuit board and a full complement of components, I've made arrangements with the folks at EDTP Electronics to provide a full kit of parts — which they call the Digital Filter Development Kit. The digital control panel application and the digital filter PIC18F452 firmware are free downloads and you can get them from the *Nuts & Volts* website at **www.nutsvolts.com** or the EDTP Electronics website at **www.edtp.com**

Note 1

Enter as many taps as you can to make your filter the best it can be. However, remember that, with an 8 kHz sampling rate, you have to process all of those taps in less than 125 mS.

Coding and Using the Digital Filter

The firmware for the digital filter is written using a combination of PIC C and PIC assembler. The PIC assembler code is used for speed. Sampling at 8 kHz means that we have 125 µS to perform the digital filter work before a new sample is taken. We want to cram every possible digital filter instruction we can between the taking of the samples. Therefore, assembler is used for coding of the math and digital filter routines.

Using PIC C allows the easy allocation of memory elements to hold the digital filter coefficients and data. The C constructs also eliminate having to code all of the house-

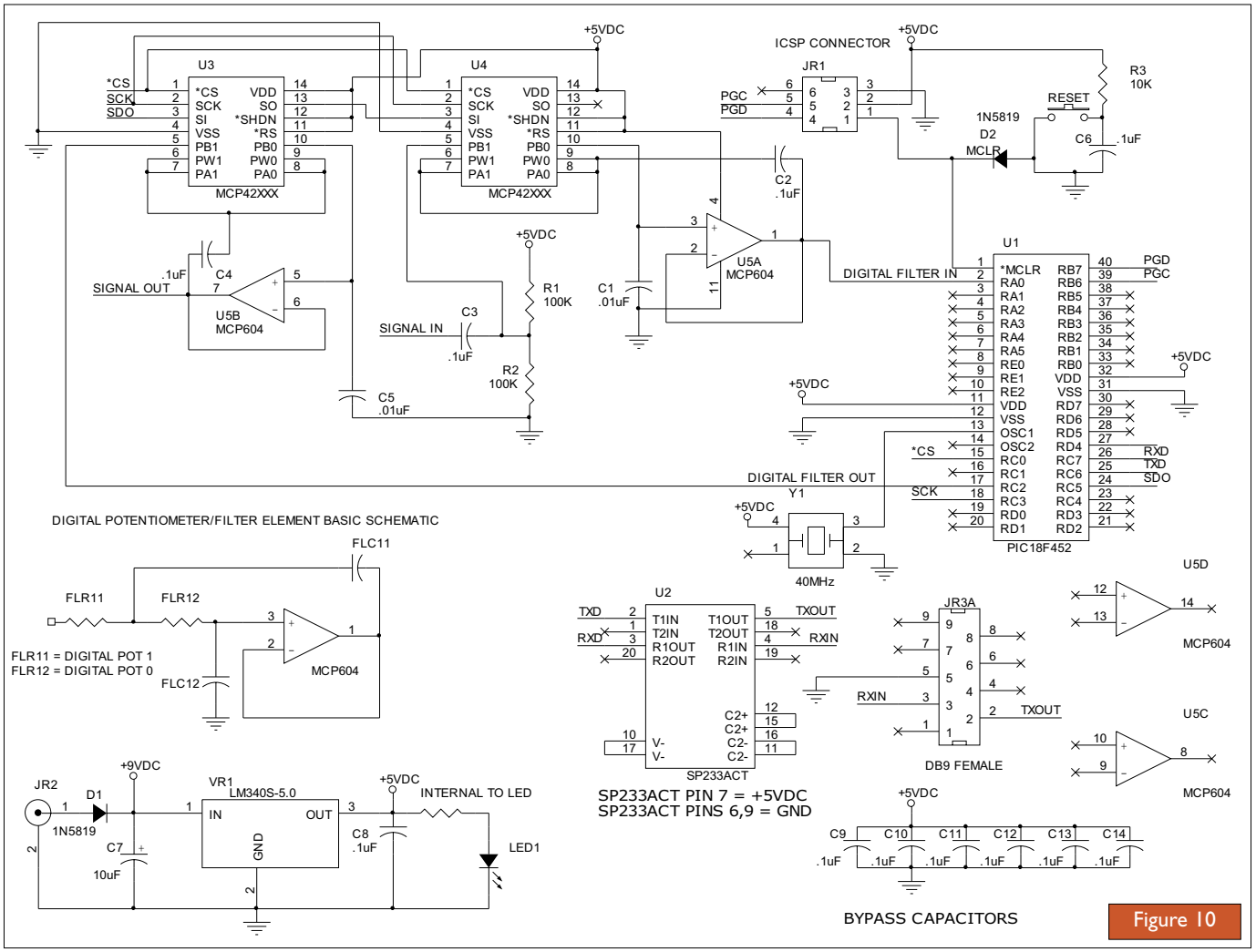


Figure 10

keeping stuff when making calls to subroutines and functions.

Arrays, constants, and variables that are defined in C can be accessed by their C-assigned names in the assembler routines. Thus, all of the PIC18F452 internal peripheral setup, the allocation of arrays and constants, and the RS-232 interrupt handler are written using C. The Custom Computer Services PIC C Compiler, in conjunction with Microchip's MPLAB IDE, is used to compile the assemble RC code combination. I used a Microchip MPLAB ICD 2 to program and debug the digital filter's PIC18F452.

The PIC18F452's Timer2 and Capture/Compare Module2's trigger special event mode are programmed to automatically kick off an analog-to-digital conversion every 125 µS. That's where the 8 kHz (1/125 µS) digital filter sample time number comes from. To avoid having to add a negative voltage reference for the PIC's analog-to-digital converter, the anti-aliasing filter's input is capacitively coupled and a +2.5 volt offset is added to the incoming signal by the voltage divider produced by R1 and R2. Doing this sets our analog-to-digital converter zero point in the center of the PIC18F452's 10-bit analog-to-digital converter range and allows the input signal to swing ±2.5 volts, which equates to 0 to 5 volts at the analog-to-digital converter input.

In order to keep input and output signals in balance, that means our digital filter output zero point must also be set for 2.5 volts. This is easily done by simply using the full 10-bit resolution of the PIC18F452's PWM module. A value of 0 x 1 FF is the offset zero point (+2.5 volts) of our analog-to-digital converter and the 50% duty cycle mark (+2.5 volts) for the PWM module. The digital filter output modulates a 39.06 kHz PWM signal that is produced by the PIC18F452's PWM module. The PIC's PWM module and the Sallen-Key filter that it is feeding form a digital-to-analog converter.

Put a voltmeter or scope probe on the output of this PWM-fed filter and you'll see +2.5 volts when the PWM duty cycle is at 50%. The ON/OFF switch inside the digital control panel's Pulse Width Modulation Manual Control box kills the automatic analog-to-digital converter process and turns off the digital filter code inside the PIC18F452. This allows the direct control of the PIC18F452's PWM module. To see how the PWM duty cycle relates to the voltage produced by the output filter, attach a voltmeter to the output of the SIGNAL OUT pin, switch on the manual control, and turn the virtual DUTY CYCLE knob. You'll see the voltage on the SIGNAL OUT pin rise and fall with the increase and decrease in the PWM duty cycle. Be sure to turn the manual PWM controls off when you're running a digital filter kernel.

Speaking of digital filter kernels, let's generate one for a low-pass FIR (Finite Impulse Response) filter using the ScopeFIR application. ScopeFIR doesn't use any of the three filter models you've been introduced to. We'll use one of those models for the analog anti-aliasing and analog digital-to-analog filter. Instead, ScopeFIR uses the

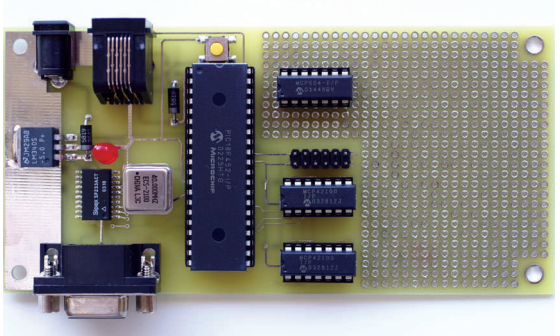
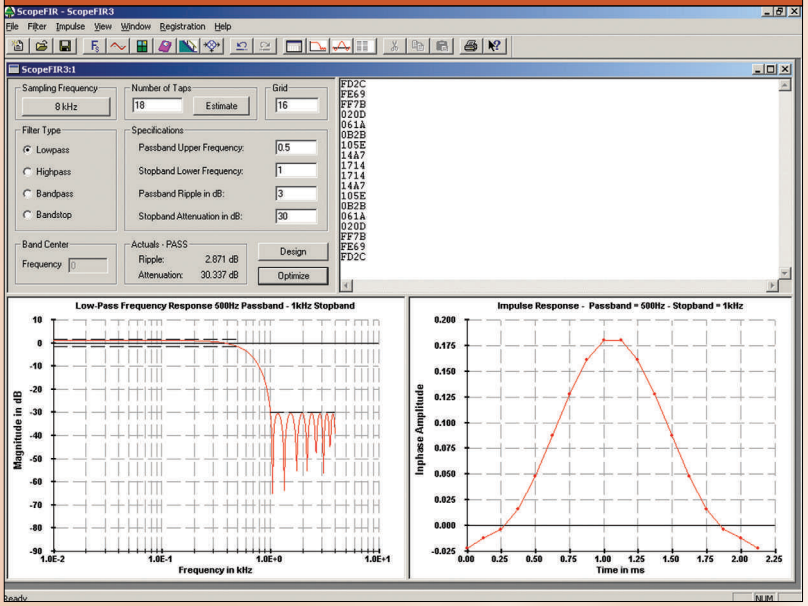


Figure 11. If you're wondering where the analog filter resistors and caps are, I breadboarded them in with 0805 SMT components on the bottom side of the board.

Parks-McClellan or Optimal Equiripple method to generate an optimized linear-phase FIR filter. Many of the free FIR filter coefficient generators that are available via the Internet use the Parks-McClellan algorithm (ScopeFIR is not free if you want the bells and whistles). Figure 12 shows a ScopeFIR session that has generated a filter kernel for a low-pass filter with a 500 Hz passband frequency and a 1 kHz stopband frequency with a -30 dB attenuation factor.

If you investigate the filter's impulse response plot points, you'll note that they are all less than 1. To make these coefficient numbers easier to handle with our assembly code math routines, we multiply — or scale — them by 32,768 or 215. If you're wondering about the negative coefficient values, they are the 16-bit hex numbers that have their most significant bit set to 1. This is called 1.15 format. That's fancy talk for 1 sign bit and 15 data bits that represent a fractional number. I instructed ScopeFIR to do the 32,768 multiplication automatically after the filter kernel was built. Once the 16-bit coefficients are determined, they

Figure 12. This is a shot of a ScopeFIR session. Everything you'll need to implement a FIR digital filter is here, including the scaled coefficients you'll need for your digital filter kernel.



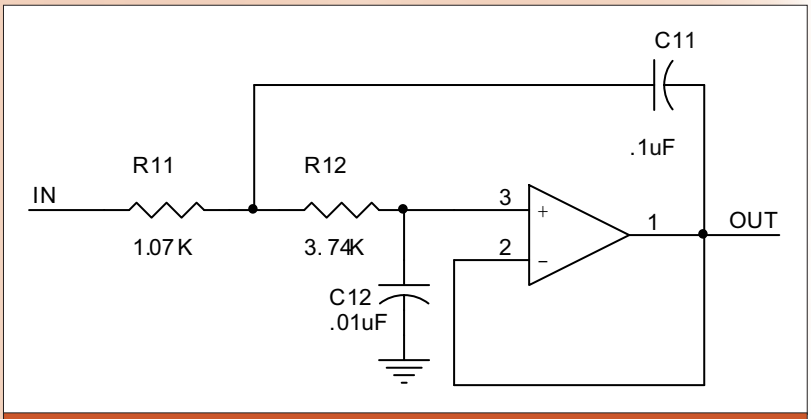


Figure 13. These values represent a Chebyshev low-pass filter with a 3 kHz bandpass and 13 kHz stopband. The stopband attenuation is set for -30 dB and the pass bandripple is set for 3 dB.

are included as constants in the digital filter firmware; more details on determining these are shown in Note 1.

Okay. Now that we have a kernel and some coefficients, we can apply them to the sum of products algorithm that form the heart of a FIR digital filter. The sum of products code implements a series of multiply and accumulate operations that follow the formula:

y[n] = a∗x[n] + a-1∗x[n-1]+ a-2∗x[n-2]+... a-j∗x[n-j] where:
y[n] = output signal
a = first coefficient
x[n] = current input signal point
a-1 = next coefficient
x[n-1] = next input signal point
until the last signal point and coefficient a-j∗x[n-j]

Each multiplication operation between the additions corresponds to a tap (coefficient) of the digital filter, which corresponds to a point in the filter kernel. The sum of products process plays the digital filter's coefficients (filter kernel) against the incoming signal. This process is called convolution. Ultimately, every incoming signal sample is convolved with every coefficient yielding a string of digitally filtered signal points.

The digitally filtered signal points are numbers that represent voltages. The digitally filtered signal points are scaled back (divided by 32,768 because we multiplied them by 32,768 earlier in the process) and applied to the PIC18F452's PWM control registers and modulate the PWM accordingly. The output analog filter is configured to eliminate the 39.06 kHz PWM signal and only allow the filtered output to flow through.

So far, we've generated a filter kernel and entered the coefficients into the PIC code. Before we can process a signal through our digital filter, we must calibrate the digital potentiometers in the digital control panel and use Filter Lab to design our anti-aliasing and PWM output analog filters.

I removed the opamp IC from the digital filter circuit and powered up the digital filter board. The Microchip digital potentiometers automatically set themselves to midpoint on powerup. I read 55 KΩ across U3 and 54.3 KΩ across U4. I chose to use a Chebyshev for both of the analog filters and set the passbands to 3 kHz with stopbands of 13 kHz with -30 dB of stopband attenuation. Since this is a Chebyshev filter, I also specified a passband ripple figure of 3 dB. Filter Lab computed the values for the Sallen-Key low-pass filters, as shown in Figure 13. I dialed all of the parameters into the digital filter control panel, as shown in Figure 14 and clicked the SET FILTERS button.

On Your Own

Wow! Figure 15 shows the signal at the output of the anti-aliasing filter (bottom sine wave) and the resultant digitally filtered signal (top sine wave). The closer I get to the 1 kHz bandstop frequency, the smaller the top sine wave gets. We've just pulled off a digital filter without having to apply any complex math. I now pass the digital filter tools over to you. **NV**

Figure 14. With the help of Microchip's Filter Lab and ScopeFIR, the digital filter control panel provides a means of quickly evaluating analog and digital filters. Note that I dialed in as close asthe potentiometers would allow.

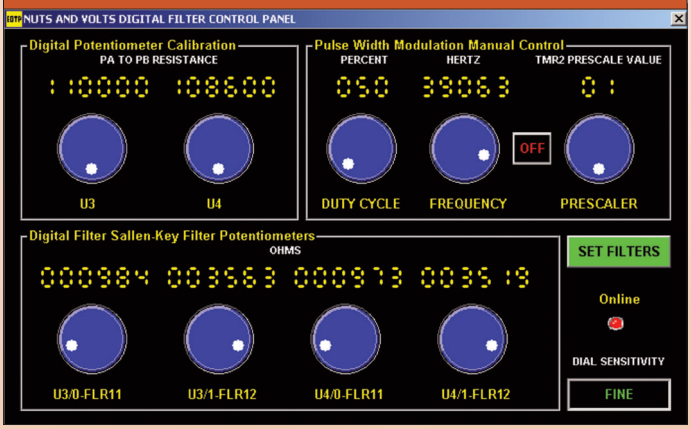


Figure 15. This photo says it all. The top sine wave's amplitude decreased as I increased the input frequency towards the digital filter's 1 kHz bandstop value.